**CHAPTER 1**

**INTRODUCTION**

**1.1 Short Description**

This Project is based on Existing Object Detection and Classification Techniques present but Instead of Classifying the Objects into any particular Language (for e.g. English ) the Android Application Developed during this project work can classify in more than one Speaking Language i.e. English,Odia,Hindi etc. The product specified belongs to the class of object detection applications but unlike other object detecting applications emphasis was put on classifying objects according to the user’s language choice in order to get more specific results. The android application is developed using android studio. For the deep learning object classifier model the concept of transfer learning is used and the mobilenet model is further trained using tensorflow deep learning environment which is then used inside the android application. The android application processes the camera feed frame by frame which is then fed to the deep net model to get the label Text. The label text is matched with the language chosen and the translated text is then displayed in the text box. If in any case the object is wrongly classified a feedback option is provided to report errors.

Upon clicking the feedback option the frame image is saved and the image along with the correct labelled text is upload to firebase cloud storage using the firebase libraries. These images will be used for further testing.

**1.2 Product Functions**

|  |  |
| --- | --- |
| **Product Functions** | **Description** |
| Scan for Objects | Look around for objects that can be classified when the user opens the application. |
| Accept the object | Get the input from the user of the object that requires to be classified |
| Process and Classify | Process the object image using the feature map inside the inbuilt Tensorflow model and return the result. |
| Translate | The Result is translated into the desired language and the result is shown to the user. |
| Error Reporting | The User reports any error made in the classification for further improvement. |

Table-1

**1.3 Operating Environment**

The Android application will require a Smartphone with an inbuilt camera. The Trained deep learning model will be kept inside the application for offline use. Error Reporting will require internet connectivity to send the reports and corrections. For Images to be clear and precise the application will require ambient light surroundings to operate on.

**1.4 Main Parts of the Application Development:-**

1. Data Collection
2. Deep Net Model Training
3. Label Translations
4. UI Development
5. Connecting Interfaces

All these steps are discussed in subsequent chapters.

**1.5 Tools Used:-**

1. **Java**

To make android application Java Programming Language is used.

Downloaded from https://www.java.com/download/

1. **Python**

For training and development of deep learning neural network python along with tensorflow is used.

Downloaded from https://www.python.org/downloads/

1. **Tensorflow**

Used for developing the Neural Network with the help of python

Installed using the command –pip install tensorflow on the command line

1. **Android Studio**

Android Studio is the official IDE to develop and debug android applications.

Downloaded from https://developer.android.com/studio

1. **A Smartphone equipped with camera**

Required to Test and Debug the Application

1. **Firebase**

Used to store the feedback images for further training and processing.



Fig-1

**Screenshot of the End Product**